CSE 211Digital Design

Akdeniz University

Week01: Introduction to Digital Design

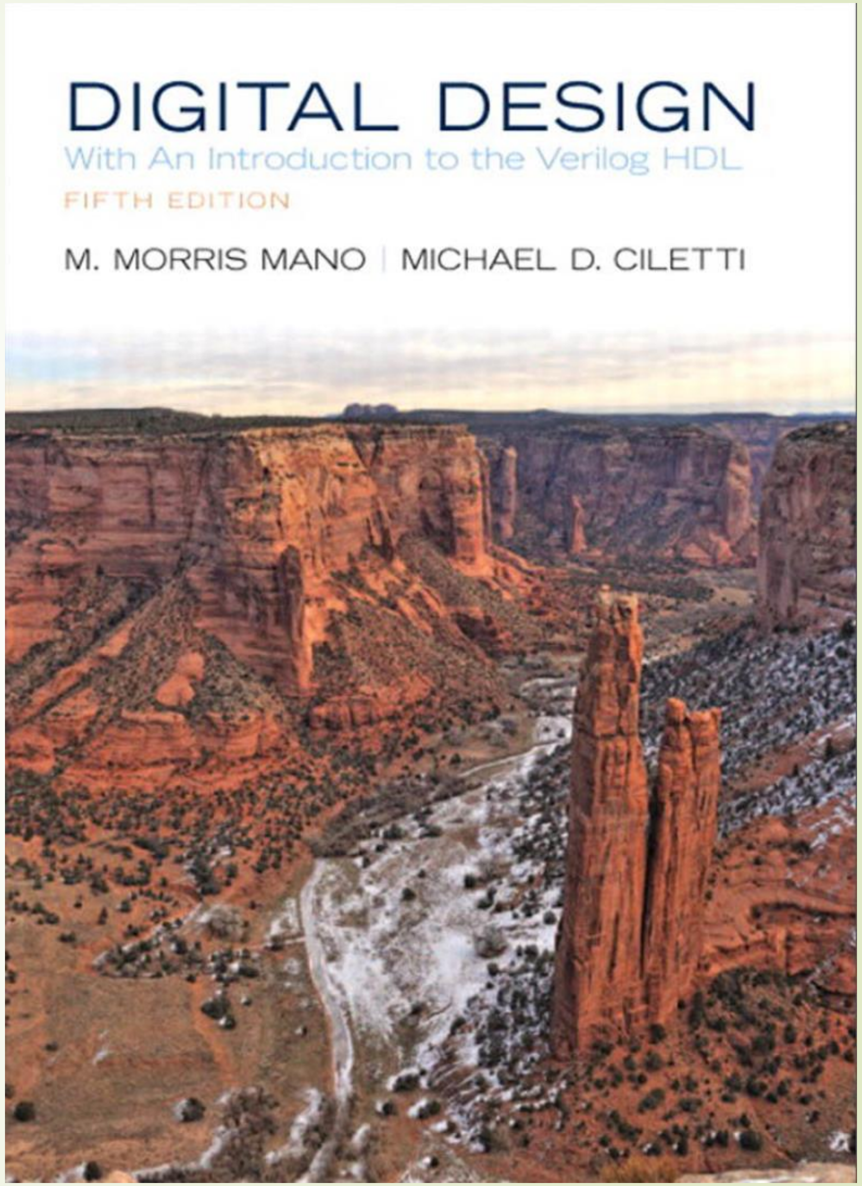
Assoc.Prof.Dr. Taner Danışman
tdanisman@akdeniz.edu.tr

Fall Semester	
16-Aug-24	Last Day for Notifying the Rectorate of Course Assignments
29-Aug-24	Last Day for Special Student Applications
September 09-13, 2024	Deadline for Payment of Contribution/Tuition Fees and Renewal of Registration
13-Sep-24	Last Day to Apply for Study Leave
16-Sep-24	Classes Begin
September 16-20, 2024	Drop and Add Period (Add-Drop)
16-Sep-24	Last Day to Notify the Rectorate of the Courses Allocated to Branches
4-Oct-24	Last Day to Withdraw from a Course
22-Dec-24	Deadline for Entering Midterm Exam Results and Other Intra-Year/Semester Measurement Tools Results into the Automation System
22-Dec-24	End of Classes
December 23, 2024 - January 03, 2025	End of Term Exams
6-Jan-25	Last Day for Entering the End of Term Exam Results into the Automation System
January 04-11, 2025	Application Dates for the Second End of Year/Semester Exam (Make-up Exam)
January 13-17, 2025	End of Year/Semester Second Exam (Make-up) Dates
20-Jan-25	Last Day for Entering the Second End of Year/Semester Exam (Supplementary Exam) Results into the Automation System

Evaluation (absolute)

- Assignments %20
- Midterm %30
- Final %50
- Teaching Assistant: Erdinç TÜRK
- Academic misconduct: **do not let this happen**
- **Monday 15:30-17:20 (Location : D206)**
- **Wednesday 15:30:17:20 (Location: D206)**

- **Textbook : Digital Design, With an Introduction to the Verilog HDL 5th Edition, Morris Mano, Michael Ciletti, Pearson.**



Can I use the previous year's Lab grades?

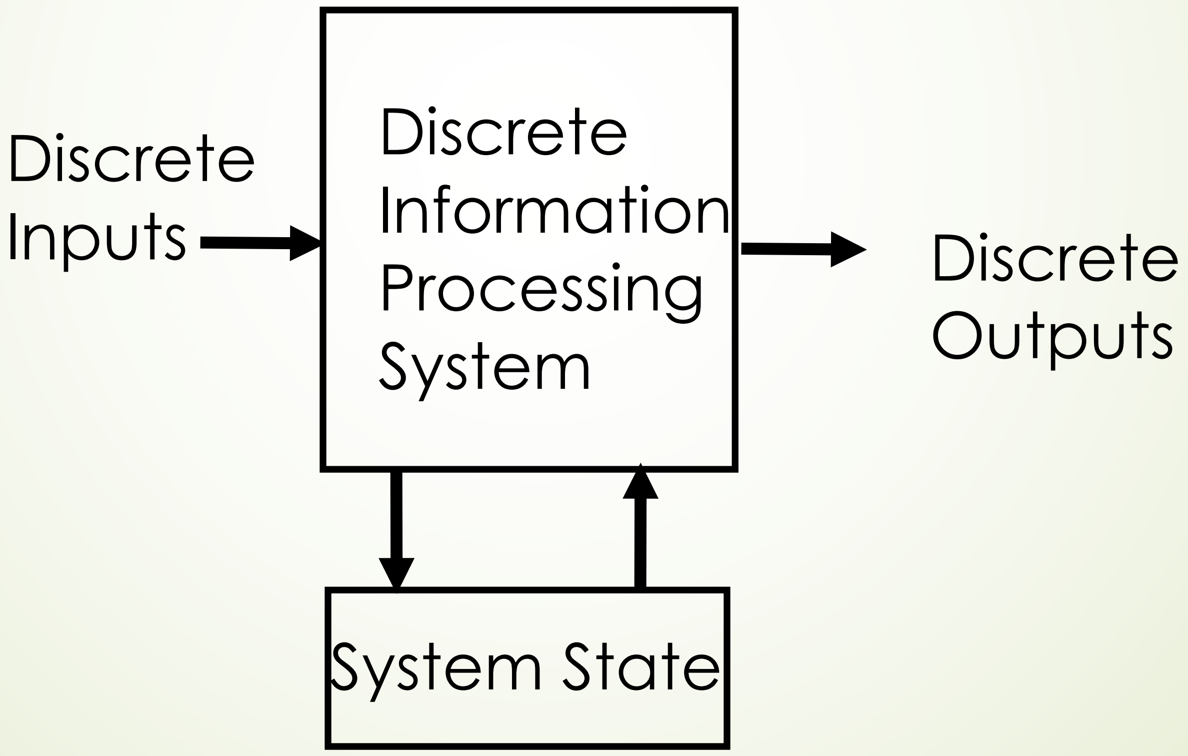
- YES, you can, but you should inform the Teaching Asistant in the beginning of the semestre (first week).

Course program

Week 01	09/16/2024	Introduction
Week 02	09/23/2024	Digital Systems and Binary Numbers I
Week 03	09/30/2024	Digital Systems and Binary Numbers II
Week 04	10/07/2024	Boolean Algebra and Logic Gates I
Week 05	10/14/2024	Boolean Algebra and Logic Gates II
Week 06	10/21/2024	Gate Level Minimization
Week 07	10/28/2024	Karnaugh Maps
Week 08	11/04/2024	Midterm
Week 09	11/11/2024	Karnaugh Maps
Week 10	11/18/2024	Combinational Logic
Week 11	11/25/2024	Combinational Logic
Week 12	12/02/2024	Timing, delays and hazards
Week 13	12/09/2024	Synchronous Sequential Logic
Week 14	12/16/2024	Synchronous Sequential Logic

Digital System

- Takes a set of discrete information <u>inputs</u> and discrete internal information <u>(system state)</u> and generates a set of discrete information <u>outputs</u>.



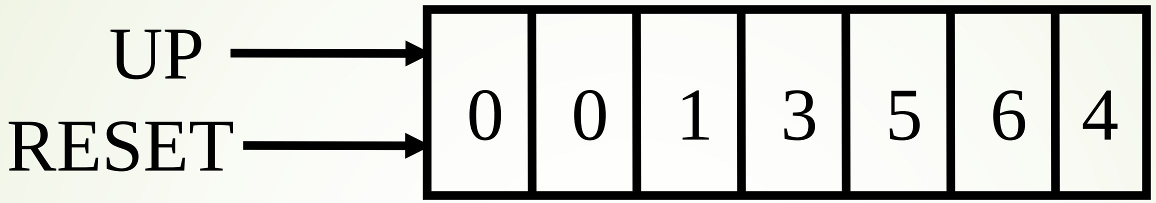
Types of Systems

- With no state present
  - Combinational logic system
  - Output = Function (Input)
- With state present
  - State updated at discrete times (e.g., once per clock tick)

    → Synchronous sequential system
  - State updated at any time

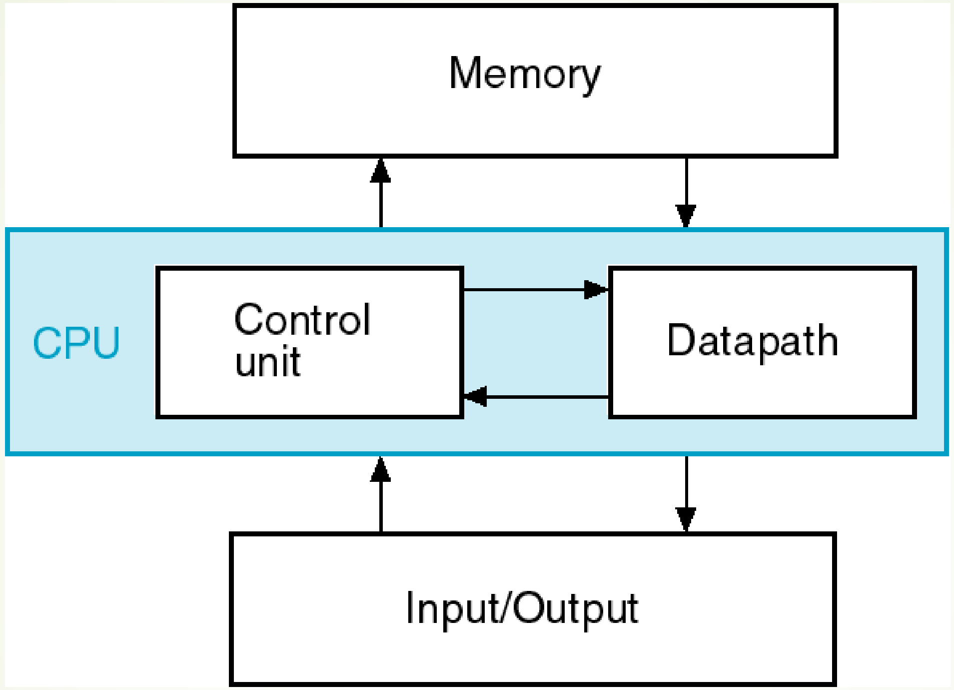
    → Asynchronous sequential system

Example: Digital Counter (e.g., Odometer)



- Inputs: Count Up, Reset
- Outputs: Visual Display
- State: “Value” of stored digits

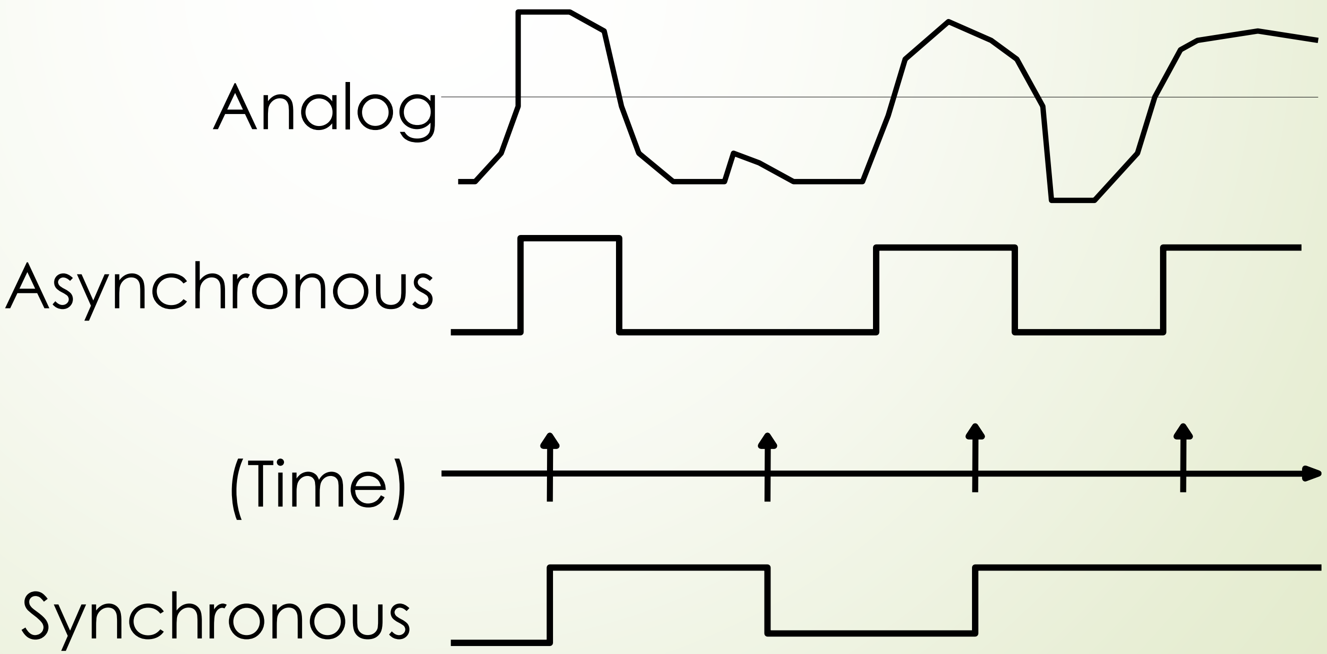
Example: Digital Computer



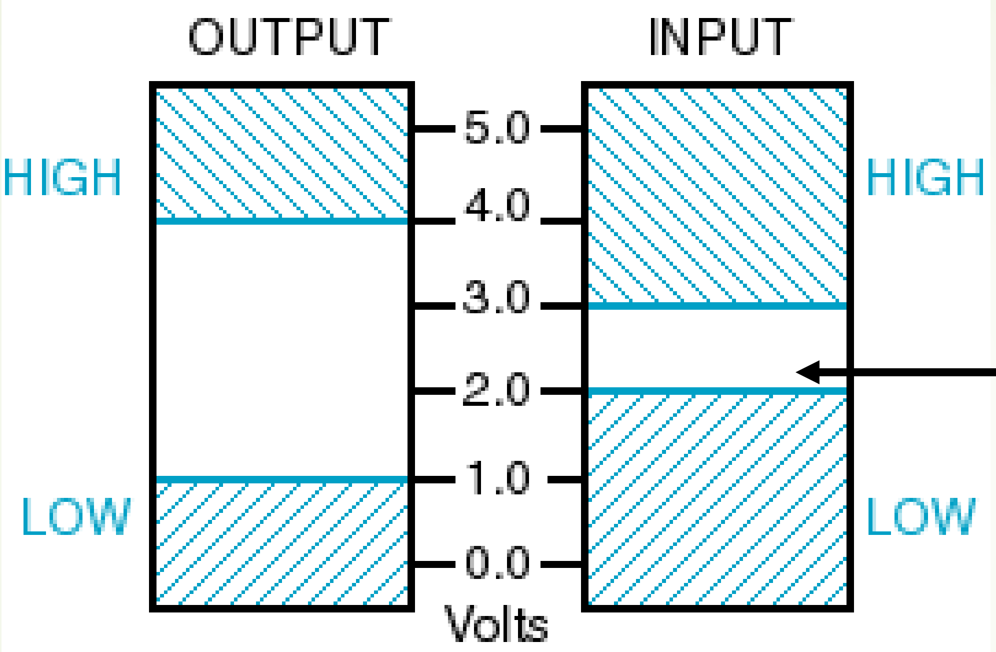
- Inputs: keyboard, mouse, modem, microphone
- Outputs: CRT, LCD, modem, speakers
- Is this system synchronous or asynchronous?

Signals

- Information variables mapped to physical quantities
- In digital systems, the quantities take on *discrete* values
  - Two-level, or *binary*, values are the most prevalent values in digital systems
  - Binary values are represented abstractly by digits 0 and 1
- Signal examples over time:



Physical Signal Example - Voltage



Threshold Region

- Other physical signals representing 1 and 0
  - CPU — Voltage
  - Disk — Magnetic field direction
  - CD — Surface pits / light
  - Dynamic RAM — Charge

Number Systems

- **Decimal Numbers**
  - What does 5,634 represent?
  - Expanding 5,634:

$$
\begin{aligned}
& 5 \times 10^3 = 5{,}000 \\
& + 6 \times 10^2 = \quad 600 \\
& + 3 \times 10^1 = \quad\ \ 30 \\
& + 4 \times 10^0 = \quad\quad 4 \quad \rightarrow 5{,}634
\end{aligned}
$$

  - What is “10” called in the above expansion?
    - The radix.
  - What is this type of number system called?
    - Decimal.
  - What are the digits for decimal numbers?
    - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
  - What are the digits for radix-r numbers?
    - 0, 1, ..., r-1.

Powers of 2

- Noteworthy powers of 2:
  - 2^{10} = kilo- = K;
  - 2^{20} = mega- = M;
  - 2^{30} = giga- = G;
  - … tera-, peta-, …

General Base Conversion

- Number Representation

Given a number of radix r of

"n" <u>integer</u> digits $a_{n-1},\ldots,a_0$

and n

"m" <u>fractional</u> digits $a_{-1},\ldots,a_{-m}$

written as:

$$a_{n-1}\ a_{n-2}\ a_{n-3}\ \ldots\ a_2\ a_1\ a_0\ .\ a_{-1}\ a_{-2}\ \ldots\ a_{-m}$$

has value:

$$(\text{Number})_r = \left(\sum_{i=0}^{i=n-1} a_i \cdot r^i\right) + \left(\sum_{j=-m}^{j=-1} a_j \cdot r^j\right)$$

(Integer Portion) + (Fraction Portion)

Commonly Occurring Bases

Name	Radix	Digits
Binary	2	0,1
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
		(= 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)

Converting Binary to Decimal

- To convert to decimal, use decimal arithmetic to sum the weighted powers of two:
- Converting 11010_2 to N_{10}:

$$N_{10} = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 26$$

Converting Decimal to Binary

- **Method 1 (Method 2 – repeated division – next slide)**
  - **Subtract the largest power of 2 that gives a positive result and record the power.**
  - **Repeat, subtracting from the prior result, until the remainder is zero.**
  - **Place 1's in the positions in the binary result corresponding to the powers recorded; in all other positions place 0's.**
- **Example: $625_{10} \rightarrow 1001110001_2$**
  - 625 – 512 = 113 → 9
  - 113 – 64 = 49 → 6
  - 49 – 32 = 17 → 5
  - 17 – 16 = 1 → 4
  - 1 – 1 = 0 → 0
  - Place 1's in the the positions recorded and 0's elsewhere
- **Converting binary to decimal:** sum weighted powers of 2 using decimal arithmetic, e.g., 512 + 64 + 32 + 16 + 1 = 625

Conversion Between Bases

- **Convert the Integral Part**
  - **Repeatedly divide the number by the radix you want to convert to and save the remainders. The new radix digits are the remainders in reverse order of computation.**

    Why does this work?

    This works because, the remainder left in the division is always

    is the coefficient of the radix's exponent.

    - **If the new radix is > 10, then convert all remainders > 10 to digits A, B, …**
- **Convert the Fractional Part**
  - **Repeatedly multiply the fraction by the radix and save the integer digits that result. The new radix fraction digits are the integer numbers in *computed order.***

    Why does this work?

    To convert fractional part, it should be divided by reciprocal of radix, which is same as multiplying with radix.

    - **If the new radix is > 10, then convert all integer numbers > 10 to digits A, B, …**
- **Join together with the radix point**

Example: Convert 46.6875_{10} To Base 2

- **Convert 46 to Base 2**

  46/2 = 23 remainder = 0

  23/2 = 11 remainder = 1

  11/2 = 5 remainder = 1

  5/2 = 2 remainder = 1

  2/2 = 1 remainder = 0

  1/2 = 0 remainder = 1

  Read off in reverse order: 101110_2

- **Convert 0.6875 to Base 2:**

  0.6875 * 2 = 1.3750 int = 1

  0.3750 * 2 = 0.7500 int = 0

  0.7500 * 2 = 1.5000 int = 1

  0.5000 * 2 = 1.0000 int = 1

  0.0000

  Read off in forward order: 0.1011_2

- **Join together with the radix point: 1011110.1011_2**

Converting Among Octal, Hexadecimal, Binary

- **Octal (Hexadecimal) to Binary:**
  - **Restate the octal (hexadecimal) as three (four) binary digits, starting at radix point and going both ways**
- **Binary to Octal (Hexadecimal):**
  - **Group the binary digits into three (four) bit groups starting at the radix point and going both ways, padding with zeros as needed in the fractional part**
  - **Convert each group of three (four) bits to an octal (hexadecimal) digit**
- **Example: Octal to Binary to Hexadecimal**

$6 \quad 3 \quad 5 \;.\; 1 \quad 7 \quad 7 \;_8$

$= 110 \,|\, 011 \,|\, 101 \;.\; 001 \,|\, 111 \,|\, 111 \;_2$

$= 1 \,|\, 1001 \,|\, 1101 \;.\; 0011 \,|\, 1111 \,|\, 1(000)_2$ (regrouping)

$= 1 \quad 9 \quad D \;.\; 3 \quad F \quad 8_{16}$ (converting)

Non-numeric Binary Codes

- **Given *n* binary digits (called <u>bits</u>), a <u>binary code</u> is a mapping from a subset of the 2^n binary numbers to some set of <u>represented elements</u>.**
- **Example: A binary code for the seven colors of the rainbow**

Binary Number	Color
000	Red
001	Orange
010	Yellow
011	Green
100	(Not mapped)
101	Blue
110	Indigo
111	Violet

- Flexibility of representation
  - can assign binary code word to any numerical or non-numerical data as long as data uniquely encoded.

Number of Bits Required

- **Given M elements to be represented by a binary code, the minimum number of bits, *n*, needed satisfies the following relationships:**
  - $2^n >= M > 2^{n-1}$
  - $n = \text{ceil}(\log_2 M)$ where ceil(x) is the smallest integer greater than or equal to x
- **Example: How many bits are required to represent <u>decimal digits</u> with a binary code?**
  - $M = 10 \rightarrow n = 4$

Number of Elements Represented

- **Given *n* digits in radix *r*, there are r^n distinct elements that can be represented.**
- **But, can represent m elements, m < r^n**
- **Examples:**
  - Can represent 4 elements in radix r = 2 with n = 2 digits: (00, 01, 10, 11)
  - Can represent 4 elements in radix r = 2 with n = 4 digits: (0001, 0010, 0100, 1000)
    - This code is called a "one hot" code

Binary Coded Decimal (BCD)

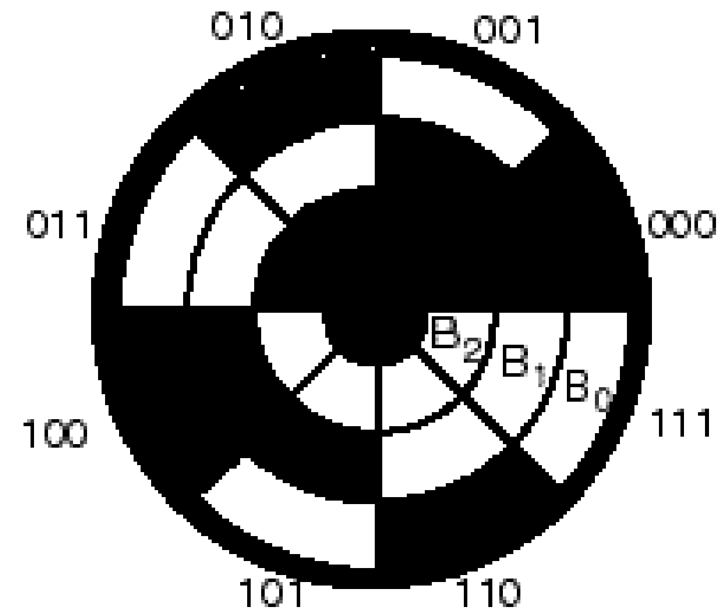
- **The BCD code is the 8,4,2,1 code.**
- **This code is the simplest, most intuitive binary code for decimal digits and uses the same weights as a binary number, but only encodes the first ten values from 0 to 9.**
- **Example: 1001 (9) = 1000 (8) + 0001 (1)**
- ***How many "invalid" code words are there?***
- ***What are the "invalid" code words?***

Gray Code

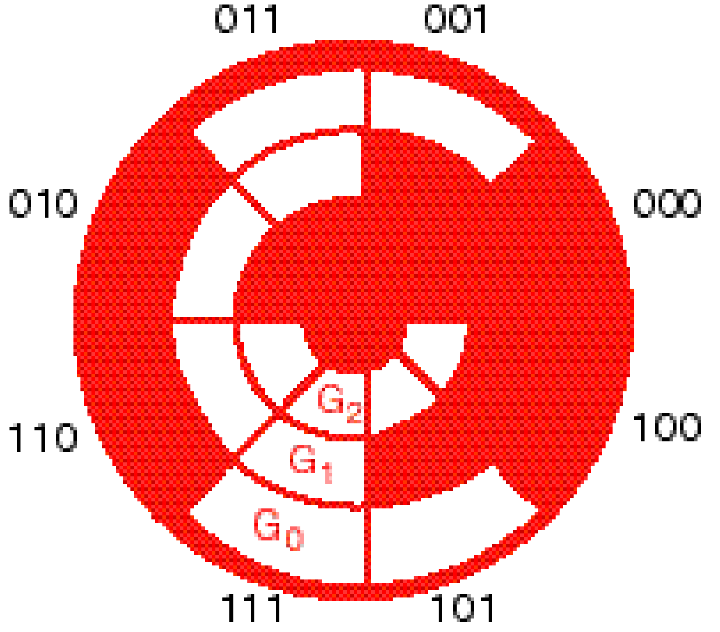
- **What property does this Gray code have?**
  - Counting up or down changes only one bit at a time (including counting between 9 and 0)

Decimal	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

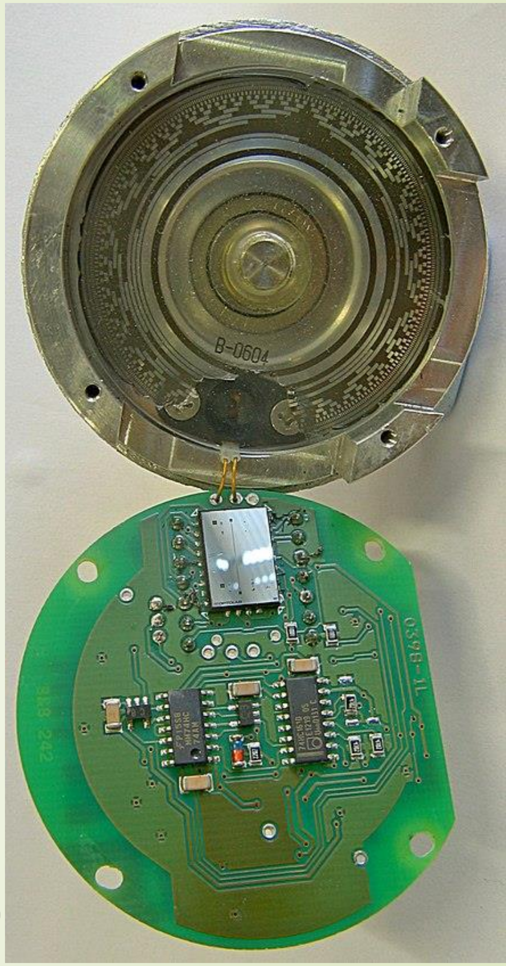
Gray Code: Optical Shaft Encoder



(a) Binary Code for Positions 0 through 7



(b) Gray Code for Positions 0 through 7



- **Shaft encoder: Capture angular position (e.g., compass)**
- **For binary code, what values can be read if the shaft position is at boundary of "3" and "4" (011 and 100) ?**
- **For Gray code, what values can be read ?**

Warning: Conversion or Coding?

- Do NOT mix up conversion of a decimal number to a binary number with coding a decimal number with a BINARY CODE.
- $13_{10} = 1101_2$ (This is conversion)
- 13 $\Leftrightarrow$ 0001|0011 (This is coding)